

Refine Search

Search Results -

Terms	Documents
L15 and stub	13


Database:

US Pre-Grant Publication Full-Text Database
 US Patents Full-Text Database
 US OCR Full-Text Database
 EPO Abstracts Database
 JPO Abstracts Database
 Derwent World Patents Index
 IBM Technical Disclosure Bulletins

Search:

L16

Refine Search

Recall Text 

Clear

Interrupt

Search History

DATE: Monday, March 03, 2008

[Purge Queries](#)[Printable Copy](#)[Create Case](#)

Set
Name Query
 side by
 side

Hit
Count
Set
Name
 result
 set

DB=USPT; PLUR=NO; OP=OR

<u>L16</u>	L15 and stub	13	<u>L16</u>
<u>L15</u>	L3 and (dead ADJ code ADJ elimination)	93	<u>L15</u>
<u>L14</u>	L3 and (stub SAME (remove or eliminate))	2	<u>L14</u>
<u>L13</u>	L3 and (stub NEAR (remove or eliminate))	0	<u>L13</u>
<u>L12</u>	L3 and (stub and (remove or eliminate))	26	<u>L12</u>
<u>L11</u>	5671419.pn.	1	<u>L11</u>
<u>L10</u>	L3 and (empty ADJ function)	7	<u>L10</u>
<u>L9</u>	L8 and (type adj analysis)	9	<u>L9</u>
<u>L8</u>	L7 or L6 or L5 or L4	157	<u>L8</u>
<u>L7</u>	L3 and instantiation	89	<u>L7</u>
<u>L6</u>	L4 and object	59	<u>L6</u>
<u>L5</u>	L4 and constraint	15	<u>L5</u>
<u>L4</u>	L3 and (exception adj handler)	74	<u>L4</u>

<u>L3</u>	(717/151 717/152 717/153 717/154 717/155 717/156 717/157 717/158 717/131 717/132 717/133).cccls.	1542	<u>L3</u>
<u>L2</u>	L1 and (exception adj handler).ab. (exception adj handler) AND ((optimization or optimizer or profiler)OR	24	<u>L2</u>
<u>L1</u>	(717/146 717/147 717/148 717/149 717/150 717/151 717/152 717/153 717/154 717/155 717/156 717/157 717/158 717/159 717/129 717/130 717/131 717/132 717/133).cccls.)	361	<u>L1</u>

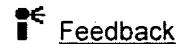
END OF SEARCH HISTORY


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide



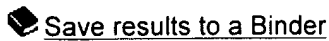
THE ACM DIGITAL LIBRARY



stub code removal

Terms used: **stub code removal**

Found 129 of 239,726

Sort results
by

 Refine these results with [Advanced Search](#)
Display
results
☐ Open results in a new
window

 Try this search in [The ACM Guide](#)

Results 1 - 20 of 129

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [next](#) [>>](#)

1 [Managing bounded code caches in dynamic binary optimization systems](#)



Kim Hazelwood, Michael D. Smith

 September 2006 **ACM Transactions on Architecture and Code Optimization (TACO)**, Volume 3 Issue 3

Publisher: ACM

 Full text available: pdf(666.72 KB) Additional Information: [full citation](#), [abstract](#),
[references](#), [index terms](#)

Dynamic binary optimizers store altered copies of original program instructions in software-managed code caches in order to maximize reuse of transformed code. Code caches store code blocks that may vary in size, reference other code blocks, and carry ...

Keywords: Dynamic optimization, code caches, dynamic translation, just-in-time compilation

2 [Removing false code dependencies to speedup software build processes](#)

Yijun Yu, Homy Dayani-Fard, John Mylopoulos

 October 2003 **CASCON '03: Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research**

Publisher: IBM Press

 Full text available: pdf(158.71 KB) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#)

The development of large software systems involves a continual lengthy build process that may include preprocessing, compilation and linking of tens of thousands of source code files. In many cases, much of this build time is wasted because of false ...

3 [Investigating the use of analysis contracts to support fault isolation in object oriented code](#)



L. C. Briand, Y. Labiche, H. Sun

Ads by Google

[Download PDF Converter](#)

 Convert Document & Image formats into PDF. Fast Download Guaranteed!
PDFConverter.PDF-forma

[Paperless Meetings, ARS](#)

 Any web-based device can access surveys, Q&A, paperless meetings!
www.visiontree.com


[Your MIS Security Career](#)

 New! Ask the MIS Security Expert. Answers to InfoSec Career Questions
www.uFairfax.net

[Decision Support Software](#)

 Demo/Purchase RightChoiceDSS for modeling critical decisions.
www.tgkconsulting.com

July 2002 **ACM SIGSOFT Software Engineering Notes**, Volume 27 Issue 4
Publisher: ACM

Full text available:  [pdf\(574.83 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#)

A number of activities involved in testing software are known to be difficult and time consuming. Among them is the isolation of faults once failures have been detected. In this paper, we investigate how the instrumentation of contracts could address ...

Keywords: contracts, object-oriented analysis, object-oriented testing, testability


4 Investigating the use of analysis contracts to support fault isolation in object oriented code



L. C. Briand, Y. Labiche, H. Sun

July 2002 **ISSTA '02**: Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis

Publisher: ACM

Full text available:  [pdf\(574.83 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#)

A number of activities involved in testing software are known to be difficult and time consuming. Among them is the isolation of faults once failures have been detected. In this paper, we investigate how the instrumentation of contracts could address ...


Keywords: contracts, object-oriented analysis, object-oriented testing, testability

5 A Cross-Architectural Interface for Code Cache Manipulation

Kim Hazelwood, Robert Cohn

March 2006 **CGO '06**: Proceedings of the International Symposium on Code Generation and Optimization

Publisher: IEEE Computer Society

Full text available:  [pdf\(407.53 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#)

Software code caches help amortize the overhead of dynamic binary transformation by enabling reuse of transformed code. Since code caches contain a potentially altered copy of every instruction that executes, run-time access to a code cache can be a very ...


6 Virgil: objects on the head of a pin



Ben L. Titzer

October 2006 **ACM SIGPLAN Notices**, Volume 41 Issue 10

Publisher: ACM

Full text available:  [pdf\(487.18 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#)

Embedded microcontrollers are becoming increasingly prolific, serving as the primary or auxiliary processor in products and research systems

from microwaves to sensor networks. Microcontrollers represent perhaps the most severely resource-constrained ...

Keywords: data-sensitive optimization, dead code elimination, embedded systems, heap compression, microcontrollers, multi-stage computation, sensor networks, standalone programs, static analysis, systems software, whole-program compilation

7 Sifting out the mud: low level C++ code reuse



Bjorn De Sutter, Bruno De Bus, Koen De Bosschere

November 2002 **OOPSLA '02**: Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications

Publisher: ACM

Full text available: [pdf\(1.35 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

More and more computers are being incorporated in devices where the available amount of memory is limited. This contrasts with the increasing need for additional functionality and the need for rapid application development. While object-oriented programming ...

Keywords: code compaction, code size reduction

8 Sifting out the mud: low level C++ code reuse



Bjorn De Sutter, Bruno De Bus, Koen De Bosschere

November 2002 **ACM SIGPLAN Notices**, Volume 37 Issue 11

Publisher: ACM

Full text available: [pdf\(1.35 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

More and more computers are being incorporated in devices where the available amount of memory is limited. This contrasts with the increasing need for additional functionality and the need for rapid application development. While object-oriented programming ...

Keywords: code compaction, code size reduction

9 Virgil: objects on the head of a pin



Ben L. Titzer

October 2006 **OOPSLA '06**: Proceedings of the 21st annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications

Publisher: ACM

Full text available: [pdf\(487.18 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Embedded microcontrollers are becoming increasingly prolific, serving as the primary or auxiliary processor in products and research systems from microwaves to sensor networks. Microcontrollers represent perhaps the most severely resource-constrained ...

Keywords: data-sensitive optimization, dead code elimination,

embedded systems, heap compression, microcontrollers, multi-stage computation, sensor networks, standalone programs, static analysis, systems software, whole-program compilation

10 HiPE on AMD64



Daniel Luna, Mikael Pettersson, Konstantinos Sagonas
September 2004 **ERLANG '04**: Proceedings of the 2004 ACM SIGPLAN
workshop on Erlang

Publisher: ACM

Full text available: [pdf\(245.37 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [index terms](#)

Erlang is a concurrent functional language designed for developing large-scale, distributed, fault-tolerant systems. The primary implementation of the language is the Erlang/OTP system from Ericsson. Even though Erlang/OTP is by default based on a virtual ...

Keywords: AMD64, erlang, native code compilation

11 DEP: detailed execution profile



Qin Zhao, Joon Edward Sim, Weng-Fai Wong, Larry Rudolph
September 2006 **PACT '06**: Proceedings of the 15th international conference
on Parallel architectures and compilation techniques

Publisher: ACM

Full text available: [pdf\(565.77 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [index terms](#)

In many areas of computer architecture design and program development, the knowledge of dynamic program behavior can be very handy. Several challenges beset the accurate and complete collection of dynamic control flow and memory reference information. ...

Keywords: control flow, dynamic instrumentation, memory reference, profile

12 Automated reduction of the memory footprint of the Linux kernel



Dominique Chanet, Bjorn De Sutter, Bruno De Bus, Ludo Van Put, Koen De Bosschere
September 2007 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 6 Issue 4

Publisher: ACM

Full text available: [pdf\(1.43 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#),
[index terms](#)

The limited built-in configurability of Linux can lead to expensive code size overhead when it is used in the embedded market. To overcome this problem, we propose the application of link-time compaction and specialization techniques that exploit the ...

Keywords: Linux kernel, compaction, compression, operating system, specialization, system calls

13 Speculative optimization using hardware-monitored guarded regions

for java virtual machines

Lixin Su, Mikko H. Lipasti

June 2007 **VEE '07**: Proceedings of the 3rd international conference on Virtual execution environments**Publisher:** ACMFull text available: [pdf\(357.43 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Aggressive dynamic optimization in high-performance Java Virtual Machines can be hampered by language features like Java's exception model, which requires precise detection and handling of program-generated exceptions. Furthermore, the compile-time overhead ...

Keywords: java, precise exceptions, speculative processors, transactional memory, virtual machines

14 Incremental and demand-driven points-to analysis using logicprogramming

Diptikalyan Saha, C. R. Ramakrishnan

July 2005 **PPDP '05**: Proceedings of the 7th ACM SIGPLAN international conference on Principles and practice of declarative programming**Publisher:** ACMFull text available: [pdf\(225.95 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Several program analysis problems can be cast elegantly as a logic program. In this paper we show how recently-developed techniques for incremental evaluation of logic programs can be refined and used for deriving practical implementations of incremental ...

Keywords: demand-drive analysis, incremental analysis, logic programming, pointer analysis

15 Runtime specialization with optimistic heap analysis

Ajeet Shankar, S. Subramanya Sastry, Rastislav Bodík, James E. Smith


October 2005 **OOPSLA '05**: Proceedings of the 20th annual ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications**Publisher:** ACMFull text available: [pdf\(425.12 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)


We describe a highly practical program specializer for Java programs. The specializer is powerful, because it specializes optimistically, using (potentially transient) constants in the heap; it is precise, because it specializes using data structures ...

Keywords: dynamic optimization, partial evaluation, program analysis, specialization

16 Hosting the .NET Runtime in Microsoft SQL server


Alazel Acheson, Mason Bendixen, José A. Blakeley, Peter Carlin, Ebru Ersan, Jun Fang, Xiaowei Jiang, Christian Kleinerman, Balaji Rathakrishnan, Gideon

-  Schaller, Beysim Sezgin, Ramachandran Venkatesh, Honggang Zhang
June 2004 **SIGMOD '04**: Proceedings of the 2004 ACM SIGMOD
international conference on Management of data
Publisher: ACM

Full text available:  [pdf\(249.27 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#)

The integration of the .NET Common Language Runtime (CLR) inside the SQL Server DBMS enables database programmers to write business logic in the form of functions, stored procedures, triggers, data types, and aggregates using modern programming languages ...

17 On reducing interprocess communication overhead in concurrent programs


-  Erik Stenman, Konstantinos Sagonas
October 2002 **ERLANG '02**: Proceedings of the 2002 ACM SIGPLAN
workshop on Erlang
Publisher: ACM


Full text available:  [pdf\(105.68 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [index terms](#)

We present several different ideas for increasing the performance of highly concurrent programs in general and Erlang programs in particular. These ideas range from simple implementation tricks that reduce communication latency to more thorough code ...

Keywords: concurrent languages, erlang, process scheduling

18 Static analysis of anomalies and security vulnerabilities in executable files

-  Jay-Evan J. Tevis, John A. Hamilton, Jr.
March 2006 **ACM-SE 44**: Proceedings of the 44th annual Southeast regional
conference
Publisher: ACM

Full text available:  [pdf\(119.85 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [index terms](#)

Software researchers have already developed static code security checkers to parse through and scan <u>source code</u> files, looking for security vulnerabilities [8, 9]. What about <u>executable</u> files? Can these files also ...

Keywords: PE format, executable file, software security vulnerabilities, static analysis

19 Run-Time Support for Optimizations Based on Escape Analysis

- Thomas Kotzmann, Hanspeter Mossenbock
March 2007 **CGO '07**: Proceedings of the International Symposium on Code
Generation and Optimization
Publisher: IEEE Computer Society

Full text available:  [pdf\(207.93 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

The Java™ programming language does not allow the programmer to influence memory management. An object is usually allocated on the heap and deallocated by the garbage collector when it is not referenced any longer. Under certain conditions, the virtual ...

20 [MJ: a rational module system for Java and its applications](#)



John Corwin, David F. Bacon, David Grove, Chet Murthy

October 2003 **OOPSLA '03**: Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications

Publisher: ACM

Full text available: [pdf\(208.83 KB\)](#)

Additional Information: [full citation](#), [abstract](#),

[references](#), [cited by](#), [index terms](#)

While Java provides many software engineering benefits, it lacks a coherent module system and instead provides only packages (which are primarily a name space mechanism) and classloaders (which are very low-level). As a result, large Java applications ...

Keywords: Java, components, language design, modularity

Results 1 - 20 of 129

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [next](#) [>>](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2008 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:



[Adobe Acrobat](#)



[QuickTime](#)



[Windows Media Player](#)



[Real Player](#)


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY

[Feedback](#)

exception handler

Terms used: **exception handler**

Found 1,500 of 239,726

Sort results by

☒ [Save results to a Binder](#)

 Refine these results with [Advanced Search](#)

Display results

☐ Open results in a new window

 Try this search in [The ACM Guide](#)

Results 1 - 20 of 1,500

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#) [>>](#)

1 [A repository of knowledge about handling exceptions in multi-agent systems](#)



Mark Klein

 May 2001 **AGENTS '01**: Proceedings of the fifth international conference on Autonomous agents

Publisher: ACM

 Full text available: [pdf\(68.50 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

A critical challenge to creating effective agent- based systems is allowing them to operate effectively in environments where failures ('exceptions') can occur. An important barrier to achieving this has been the lack of systematized dissemination of ...

Keywords: exception handling knowledge base multi-agent systems

[Ads by Google](#)

[Download PDF Converter](#)

Convert Document & Image formats into PDF. Fast Download Guaranteed!
PDFConverter.PDF-forma

[Information Management](#)

Trust SAS to manage your data & information- Get Info Now!
[www.SAS.com](#)

[Paperless Meetings, ARS](#)

Any web-based device can access surveys, Q&A, paperless meetings!
[www.visiontree.com](#)

[Your MIS Security Career](#)

New! Ask the MIS Security Expert. Answers to InfoSec Career Questions
[www.uFairfax.net](#)

2 [Modeling exceptions via commitment protocols](#)



Ashok U. Mallya, Munindar P. Singh

 July 2005 **AAMAS '05**: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems

Publisher: ACM

 Full text available: [pdf\(336.93 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

This paper develops a model for exceptions and an approach for incorporating them in commitment protocols among autonomous agents. Modeling and handling exceptions is critical for successful applications of multiagent systems. Protocols help build multiagent ...

Keywords: agents, commitments, exception handling, multiagent systems

3 [Exception handling in workflow-driven Web applications](#)



Marco Brambilla, Stefano Ceri, Sara Comai, Christina Tziviskou

 May 2005 **WWW '05**: Proceedings of the 14th international conference on World Wide Web

Publisher: ACM

Full text available:  [pdf\(259.54 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [index terms](#)

As the Web becomes a platform for implementing B2B applications, the need arises of Web conceptual models for describing Web oriented workflow applications implementing business processes. In this context, new problems about process correctness arise, ...


Keywords: Web applications, exceptions, failure, navigation behavior, workflow

4 A study of exception handling and its dynamic optimization in Java



Takeshi Ogasawara, Hideaki Komatsu, Toshio Nakatani
November 2001 **ACM SIGPLAN Notices**, Volume 36 Issue 11

Publisher: ACM

Full text available:  [pdf\(190.18 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#)


Optimizing exception handling is critical for programs that frequently throw exceptions. We observed that there are many such exception-intensive programs in various categories of Java programs. There are two commonly used exception handling techniques, ...

5 A framework for analyzing exception flow in software architectures



Fernando Castor Filho, Patrick H. S. Brito, Cecília Mary F. Rubira
May 2005 **WADS '05: Proceedings of the 2005 workshop on Architecting dependable systems**

Publisher: ACM

Full text available:  [pdf\(194.49 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#)

We present Aereal, a framework for analyzing exception flow in architecture descriptions. Aereal works as a customizable architectural-level exception handling system that can be further constrained or have some of its rules relaxed. Since different ...


Keywords: architecture analysis, architecture documentation, exception handling

6 Exceptions and aspects: the devil is in the details



Fernando Castor Filho, Nelio Cacho, Eduardo Figueiredo, Raquel Maranhão, Alessandro Garcia, Cecília Mary F. Rubira
November 2006 **SIGSOFT '06/FSE-14: Proceedings of the 14th ACM SIGSOFT international symposium on Foundations of software engineering**

Publisher: ACM

Full text available:  [pdf\(354.17 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#)

It is usually assumed that the implementation of exception handling can

be better modularized by the use of aspect-oriented programming (AOP). However, the trade-offs involved in using AOP with this goal are not well-understood. This paper presents an ...

Keywords: aspectJ, empirical studies, exception handling

7 Optimization and precise exceptions in dynamic compilation



Michael Gschwind, Erik Altman

March 2001 **ACM SIGARCH Computer Architecture News**, Volume 29 Issue 1

Publisher: ACM

Full text available: pdf(508.52 KB) Additional Information: [full citation](#), [abstract](#), [index terms](#)

Maintaining precise exceptions is an important aspect of achieving full compatibility with a legacy architecture. While asynchronous exceptions can be deferred to an appropriate boundary in the code, synchronous exceptions must be taken when they occur. ...

8 Static analysis to support the evolution of exception structure in object-oriented systems



Martin P. Robillard, Gail C. Murphy

April 2003 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 12 Issue 2

Publisher: ACM

Full text available: pdf(708.48 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#), [review](#)

Exception-handling mechanisms in modern programming languages provide a means to help software developers build robust applications by separating the normal control flow of a program from the control flow of the program under exceptional situations. ...

Keywords: Error handling, exception flow, exception handling, exception structure, program evolution, static analysis

9 Exception analysis for non-strict languages



Kevin Glynn, Peter J. Stuckey, Martin Sulzmann, Harald Søndergaard

September 2002 **ACM SIGPLAN Notices**, Volume 37 Issue 9

Publisher: ACM

Full text available: pdf(241.32 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

In this paper we present the first exception analysis for a non-strict language. We augment a simply-typed functional language with exceptions, and show that we can define a type-based inference system to detect uncaught exceptions. We have implemented ...

Keywords: Boolean constraints, effect systems, exceptions, non-strict functional programming languages, type inference

10 An efficient and reliable object-oriented exception handling mechanism



Shujuan Jiang, Baowen Xu

February 2005 **ACM SIGPLAN Notices**, Volume 40 Issue 2

Publisher: ACM

Full text available: [pdf\(888.84 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [index terms](#)

This paper proposes an exception handling mechanism for developing reliable object-oriented systems based on analyzing some problems encountered in the C++ programming language. The exceptions are organized into a knowledge sharing inheritance hierarchy ...

Keywords: C++, exception handling, inheritance hierarchy, object-oriented systems, programming languages

11 Exception analysis for non-strict languages



Kevin Glynn, Peter J. Stuckey, Martin Sulzmann, Harald Søndergaard

October 2002 **ICFP '02: Proceedings of the seventh ACM SIGPLAN international conference on Functional programming**

Publisher: ACM

Full text available: [pdf\(241.32 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#)

In this paper we present the first exception analysis for a non-strict language. We augment a simply-typed functional language with exceptions, and show that we can define a type-based inference system to detect uncaught exceptions. We have implemented ...

Keywords: Boolean constraints, effect systems, exceptions, non-strict functional programming languages, type inference

12 MOPping up exceptions



S. E. Mitchell, A. Burns, A. J. Wellings

September 2001 **ACM SIGAda Ada Letters**, Volume XXI Issue 3

Publisher: ACM

Full text available: [pdf\(924.06 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#)

This paper describes the development of a model for the reflective treatment of both application and environmentally sourced exceptions. We show how a variety of exception models can be implemented using an exception handler at the metalevel. The approach ...

Keywords: exceptions, metalevel architecture, reflection


13 Implementing the complex arcsine and arccosine functions using exception handling



T. E. Hull, Thomas F. Fairgrieve, Ping Tak Peter Tang

September 1997 **ACM Transactions on Mathematical Software (TOMS)**, Volume 23 Issue 3

Publisher: ACM

Full text available:  [pdf\(310.36 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#)

We develop efficient algorithms for reliable and accurate evaluations of the complex arcsine and arccosine functions. A tight error bound is derived for each algorithm; the results are valid for all machine-representable points in the complex plane. The ...

Keywords: complex elementary functions, implementation


14 Ada exception handling: an axiomatic approach



David C. Luckham, W. Polak

April 1980 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 2 Issue 2

Publisher: ACM

Full text available:  [pdf\(481.97 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#)

A method of documenting exception propagation and handling in Ada programs is proposed. Exception propagation declarations are introduced as a new component of Ada specifications, permitting documentation of those exceptions that can be propagated by ...


15 A study of the applicability of existing exception-handling techniques to component-based real-time software technology



Jun Lang, David B. Stewart

March 1998 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 20 Issue 2

Publisher: ACM

Full text available:  [pdf\(220.57 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#)

This study focuses on the current state of error-handling technology and concludes with recommendations for further research in error handling for component-based real-time software. With real-time programs growing in size and complexity, the quality ...

Keywords: component-based software, error detection and handling, faults, reconfigurable software, signals, survey, timing and deadline failures


16 Language features for flexible handling of exceptions in information systems



Alexander Borgida

December 1985 **ACM Transactions on Database Systems (TODS)**, Volume 10 Issue 4

Publisher: ACM

Full text available:  [pdf\(3.12 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#),
[cited by](#), [index terms](#), [review](#)

An exception-handling facility suitable for languages used to implement

database-intensive information systems is presented. Such a mechanism facilitates the development and maintenance of more flexible software systems by supporting the abstraction ...

17 Workflow as persistent objects with persistent exceptions: a framework for flexibility



Alex Borgida, Takahiro Murata

December 1999 **ACM SIGGROUP Bulletin**, Volume 20 Issue 3

Publisher: ACM

Full text available: [pdf\(196.72 KB\)](#) Additional Information: [full citation](#), [abstract](#)

It is of significant value for an organization to be able to analyze and assist business processes by capturing them in a *process modeling language*. It describes the tasks to be performed in steps and their coordination in a *schema*, ...

18 A modular verifiable exception handling mechanism



Shaula Yemini, Daniel M. Berry

April 1985 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 7 Issue 2

Publisher: ACM

Full text available: [pdf\(2.38 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#), [review](#)

This paper presents a new model for exception handling, called the replacement model. The replacement model, in contrast to other exception-handling proposals, supports all the handler responses of resumption, termination, retry, and exception propagation, ...

19 Hardware and software support for efficient exception handling



Chandramohan A. Thekkath, Henry M. Levy

November 1994 **ACM SIGPLAN Notices**, Volume 29 Issue 11

Publisher: ACM

Full text available: [pdf\(1.44 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Program-synchronous exceptions, for example, breakpoints, watchpoints, illegal opcodes, and memory access violations, provide information about exceptional conditions, interrupting the program and vectoring to an operating system handler. ...

20 Exception handling in APL



Dennis R. Adler

July 1982 **APL '82**: Proceedings of the international conference on APL

Publisher: ACM

Full text available: [pdf\(455.80 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

This paper examines APL exception handling facilities as they relate to applications programming. A brief background on exception handling is first presented. Next, the qualities most desirable in an exception handler are discussed. These criteria are ...

Results 1 - 20 of 1,500 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#) [>>](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2008 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

Freeform Search

Database:	<div style="border: 1px solid black; padding: 2px;"> US Pre-Grant Publication Full-Text Database US Patents Full-Text Database US OCR Full-Text Database EPO Abstracts Database JPO Abstracts Database Derwent World Patents Index IBM Technical Disclosure Bulletins </div>
Term:	<div style="border: 1px solid black; padding: 2px;"> L18 and folding </div>
Display:	<div style="border: 1px solid black; padding: 2px; display: inline-block;">50</div> Documents in Display Format: <div style="border: 1px solid black; padding: 2px; display: inline-block;">REV</div> Starting with Number <div style="border: 1px solid black; padding: 2px; display: inline-block;">1</div>
Generate: <input type="radio"/> Hit List <input checked="" type="radio"/> Hit Count <input type="radio"/> Side by Side <input type="radio"/> Image	

Search

Clear

Interrupt

Search History

DATE: Monday, March 03, 2008
 [Purge Queries](#)
 [Printable Copy](#)
 [Create Case](#)

Set Name	Query	Hit Count	Set Name result set
side by side			
<i>DB=USPT; PLUR=NO; OP=OR</i>			
<u>L19</u>	L18 and folding	12	<u>L19</u>
<u>L18</u>	L17 and elimination	76	<u>L18</u>
<u>L17</u>	L3 and (compare)	312	<u>L17</u>
<u>L16</u>	L15 and stub	13	<u>L16</u>
<u>L15</u>	L3 and (dead ADJ code ADJ elimination)	93	<u>L15</u>
<u>L14</u>	L3 and (stub SAME (remove or eliminate))	2	<u>L14</u>
<u>L13</u>	L3 and (stub NEAR (remove or eliminate))	0	<u>L13</u>
<u>L12</u>	L3 and (stub and (remove or eliminate))	26	<u>L12</u>
<u>L11</u>	5671419.pn.	1	<u>L11</u>
<u>L10</u>	L3 and (empty ADJ function)	7	<u>L10</u>
<u>L9</u>	L8 and (type adj analysis)	9	<u>L9</u>
<u>L8</u>	L7 or L6 or L5 or L4	157	<u>L8</u>
<u>L7</u>	L3 and instantiation	89	<u>L7</u>
<u>L6</u>	L4 and object	59	<u>L6</u>
<u>L5</u>	L4 and constraint	15	<u>L5</u>

<u>L4</u>	L3 and (exception adj handler)	74	<u>L4</u>
<u>L3</u>	(717/151 717/152 717/153 717/154 717/155 717/156 717/157 717/158 717/131 717/132 717/133).cccls.	1542	<u>L3</u>
<u>L2</u>	L1 and (exception adj handler).ab. (exception adj handler) AND ((optimization or optimizer or profiler)OR	24	<u>L2</u>
<u>L1</u>	(717/146 717/147 717/148 717/149 717/150 717/151 717/152 717/153 717/154 717/155 717/156 717/157 717/158 717/159 717/129 717/130 717/131 717/132 717/133).cccls.)	361	<u>L1</u>

END OF SEARCH HISTORY

Print Selection

Section: Page(s): Print Copy:

Print? <input type="button" value="All"/> <input type="button" value="Clr"/>	Change? <input type="button" value="All"/> <input type="button" value="Clr"/>	Document ID	Section(s)	Page# or Page Range to Print	Total # of Pages	# of Copies	Database
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5836014	<input type="text" value="all"/> <input type="button" value="v"/>	<input type="text" value="all"/>	* 72	<input type="text" value="1"/> <input type="button" value="v"/>	USPT

Note: Print requests for more than 49 pages are denoted by '*' and are in red.

Building	Room	Printer
<input type="text" value="ran"/> <input type="button" value="v"/>	<input type="text" value="5c18"/> <input type="button" value="v"/>	<input type="text" value="gbrgptr"/> <input type="button" value="v"/>

[Web](#) [Images](#) [Maps](#) [News](#) [Shopping](#) [Gmail](#) [more](#) ▾

[Sign in](#)

[Google](#)

Fast Static Analysis of C++ Virtual Function C

[Search](#)

[Advanced Search](#)
[Preferences](#)

Web Results 1 - 10 of about 177 for **Fast Static Analysis of C++ Virtual Function Calls, David F. Bacon et al,**

Fast Static Analysis of C++ Virtual Function Calls - Bacon ...

David F. Bacon and Peter F. Sweeney. **Fast Static Analysis of C++ Virtual Function ... 7**
virtual function calls (context) - Porat, Bernstein et al. - 1996 ...
citeseer.ist.psu.edu/276967.html - 26k - [Cached](#) - [Similar pages](#)

Fast Static Analysis of C++ Virtual Function Calls

Fast Static Analysis of C++ Virtual Function Calls. David F. Bacon and Peter F. Sweeney Dean et al. [11] studied **virtual method call elimina-** ...
portal.acm.org/ft_gateway.cfm?id=236371&
type=pdf&coll=&dl=&CFID=15151515&CFTOKEN=6184618 - [Similar pages](#)

Static analysis for dynamic coupling measures

Fast static analysis of C++ virtual function calls. In ACM Conference on ... Programming Systems, Languages, and Applications, **pages 324--341, 1996.** ...
portal.acm.org/citation.cfm?id=1188980 - [Similar pages](#)
[More results from portal.acm.org »](#)

Optimizing safe downcasting in an object-oriented programming ...

Fast Static Analysis of C++ Virtual Function Calls, David f. Bacon et al, IBM, ACM, .
1996, pp. 324-341. "Coping with Type Casts in C", Michael Siff et al, ...
www.freepatentsonline.com/7171649.html - 52k - [Cached](#) - [Similar pages](#)

Introduction

Fast static analysis of C++ virtual function calls. In Object-Oriented Programming Systems, Languages and Applications '96, **pages. 324-341.** ACM Press, 1996. ...
www.springerlink.com/index/r53r521837r25017.pdf - [Similar pages](#)

Fast Escape Analysis and Stack Allocation for Object-Based Programs

David F. Bacon and Peter F. Sweeney. **Fast static analysis of C++ virtual. function calls.**
... SIGPLAN Notices, **pages 324--341**, New York, October6--10 1996. ACM ...
www.springerlink.com/index/C5KPMHL6NMH1RMMG.pdf - [Similar pages](#)
[More results from www.springerlink.com »](#)

[PS] An overview of Types in Compilation

File Format: Adobe PostScript - [View as Text](#)

David Bacon and Peter Sweeney. **Fast static analysis of C++ virtual function calls. ...**
pages. 324{341. ACM Press, 1996. 6. Henry G. Baker. ...
gallium.inria.fr/~xleroy/publi/intro-tic98.ps.gz - [Similar pages](#)

A Low-Level Analysis Library for Architecture Recovery

[4] David F. Bacon and Peter F. Sweeney. **Fast static analysis of C++ virtual. function**
calls. ACM SIGPLAN Notices, 31(10):**324--341, 1996.** ...
linkinghub.elsevier.com/retrieve/pii/S1571066104807369 - [Similar pages](#)

A generic worklis algorithm for graph reachability problems in ...

[2] David F. Bacon and Peter F. Sweeney. **Fast static anal. ysis of C++ virtual function**
calls. In Coplien [6],. **pages 324 - 341.** ...
ieeexplore.ieee.org/iel5/7817/21484/00995791.pdf - [Similar pages](#)

[PDF] [Marmot: An Optimizing Compiler for Java](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

David F. Bacon and **Peter F. Sweeney**. **Fast static analysis of C++ virtual function calls.**

In Proceedings. OOPSLA '96, **ACM SIGPLAN Notices**, pages **324–341**, ...

research.microsoft.com/~dtarditi/dist/marmot.pdf - [Similar pages](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) **Next**

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#) | [Try Google Experimental](#)

©2008 Google - [Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

Refine Search

Search Results -

Terms	Documents
L29 and native.ab.	8

Database:

US Pre-Grant Publication Full-Text Database
 US Patents Full-Text Database
 US OCR Full-Text Database
 EPO Abstracts Database
 JPO Abstracts Database
 Derwent World Patents Index
 IBM Technical Disclosure Bulletins

Search:

L30

Refine Search

Recall Text

Clear

Interrupt

Search History

DATE: Monday, March 03, 2008

[Purge Queries](#)[Printable Copy](#)[Create Case](#)

Set
 Name Query
 side by
 side

Hit
 Count Set
 Name
 result
 set

DB=USPT; PLUR=NO; OP=OR

<u>L30</u>	L29 and native.ab.	8	<u>L30</u>
<u>L29</u>	L27 and 717/151-159.ccls.	52	<u>L29</u>
<u>L28</u>	L27 and 717/\$\$\$ccls.	192	<u>L28</u>
<u>L27</u>	L26 and (virtual ADJ function)	453	<u>L27</u>
<u>L26</u>	(source AND Code and optimization) and native (object same type)	290844	<u>L26</u>
<u>L25</u>	L24 and constant	12	<u>L25</u>
<u>L24</u>	L17 and (Loop).ab.	23	<u>L24</u>
<u>L23</u>	L18 and folding.ab.	2	<u>L23</u>
<u>L22</u>	L17 and (Loop and constant).ab.	0	<u>L22</u>
<u>L21</u>	L20 and (eliminate or replace)	62	<u>L21</u>
<u>L20</u>	L17 and Loop and constant	91	<u>L20</u>
<u>L19</u>	L18 and folding	12	<u>L19</u>
<u>L18</u>	L17 and elimination	76	<u>L18</u>

<u>L17</u>	L3 and (compare)	312	<u>L17</u>
<u>L16</u>	L15 and stub	13	<u>L16</u>
<u>L15</u>	L3 and (dead ADJ code ADJ elimination)	93	<u>L15</u>
<u>L14</u>	L3 and (stub SAME (remove or eliminate))	2	<u>L14</u>
<u>L13</u>	L3 and (stub NEAR (remove or eliminate))	0	<u>L13</u>
<u>L12</u>	L3 and (stub and (remove or eliminate))	26	<u>L12</u>
<u>L11</u>	5671419.pn.	1	<u>L11</u>
<u>L10</u>	L3 and (empty ADJ function)	7	<u>L10</u>
<u>L9</u>	L8 and (type adj analysis)	9	<u>L9</u>
<u>L8</u>	L7 or L6 or L5 or L4	157	<u>L8</u>
<u>L7</u>	L3 and instantiation	89	<u>L7</u>
<u>L6</u>	L4 and object	59	<u>L6</u>
<u>L5</u>	L4 and constraint	15	<u>L5</u>
<u>L4</u>	L3 and (exception adj handler)	74	<u>L4</u>
<u>L3</u>	(717/151 717/152 717/153 717/154 717/155 717/156 717/157 717/158 717/131 717/132 717/133).ccls.	1542	<u>L3</u>
<u>L2</u>	L1 and (exception adj handler).ab.	24	<u>L2</u>
<u>L1</u>	(exception adj handler) AND ((optimization or optimizer or profiler)OR (717/146 717/147 717/148 717/149 717/150 717/151 717/152 717/153 717/154 717/155 717/156 717/157 717/158 717/159 717/129 717/130 717/131 717/132 717/133).ccls.)	361	<u>L1</u>

END OF SEARCH HISTORY

Print Selection

Section:
 Page(s):
 Print Copy:

Print? <input type="button" value="All"/> <input type="button" value="Clr"/>	Change? <input type="button" value="All"/> <input type="button" value="Clr"/>	Document ID	Section(s)	Page# or Page Range to Print	Total # of Pages	# of Copies	Database
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6226789	<input type="text" value="all"/> <input type="button" value="v"/>	<input type="text" value="all"/>	* 158	<input type="text" value="1"/> <input type="button" value="v"/>	USPT
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6091897	<input type="text" value="all"/> <input type="button" value="v"/>	<input type="text" value="all"/>	* 158	<input type="text" value="1"/> <input type="button" value="v"/>	USPT
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5842017	<input type="text" value="all"/> <input type="button" value="v"/>	<input type="text" value="all"/>	* 160	<input type="text" value="1"/> <input type="button" value="v"/>	USPT
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5930509	<input type="text" value="all"/> <input type="button" value="v"/>	<input type="text" value="all"/>	* 119	<input type="text" value="1"/> <input type="button" value="v"/>	USPT

Note: Print requests for more than 49 pages are denoted by '*' and are in red.

Building **Room** **Printer**

Freeform Search

Database:	<div style="border: 1px solid black; padding: 2px;"> US Pre-Grant Publication Full-Text Database US Patents Full-Text Database US OCR Full-Text Database EPO Abstracts Database JPO Abstracts Database Derwent World Patents Index IBM Technical Disclosure Bulletins </div>
Term:	<div style="border: 1px solid black; padding: 2px;"> L35 AND 717/150-159.ccls. </div>
Display:	<div style="border: 1px solid black; padding: 2px; display: inline-block;">50</div> Documents in Display Format: <div style="border: 1px solid black; padding: 2px; display: inline-block;">REV</div> Starting with Number <div style="border: 1px solid black; padding: 2px; display: inline-block;">1</div>
Generate: <input type="radio"/> Hit List <input checked="" type="radio"/> Hit Count <input type="radio"/> Side by Side <input type="radio"/> Image	

Search

Clear

Interrupt

Search History

DATE: Monday, March 03, 2008
 [Purge Queries](#)
 [Printable Copy](#)
 [Create Case](#)

<u>Set</u> <u>Name</u>	<u>Query</u>	<u>Hit</u> <u>Count</u>	<u>Set</u> <u>Name</u> result set
side by side			
<i>DB=USPT; PLUR=NO; OP=OR</i>			
<u>L36</u>	L35 AND 717/150-159.ccls.	25	<u>L36</u>
<u>L35</u>	L33 or l32	158	<u>L35</u>
<u>L34</u>	L33 and l32	0	<u>L34</u>
<u>L33</u>	native AND type AND object AND (virtual ADJ function)	151	<u>L33</u>
<u>L32</u>	virtual AND function AND constraints AND (sensitive ADJ analysis)	7	<u>L32</u>
<u>L31</u>	native AND type AND object AND virtual AND function AND constraints AND (sensitive ADJ analysis)	0	<u>L31</u>
<u>L30</u>	L29 and native.ab.	8	<u>L30</u>
<u>L29</u>	L27 and 717/151-159.ccls.	52	<u>L29</u>
<u>L28</u>	L27 and 717/\$\$\$ccls.	192	<u>L28</u>
<u>L27</u>	L26 and (virtual ADJ function)	453	<u>L27</u>
<u>L26</u>	(source AND Code and optimization) and native (object same type)	290844	<u>L26</u>
<u>L25</u>	L24 and constant	12	<u>L25</u>
<u>L24</u>	L17 and (Loop).ab.	23	<u>L24</u>
<u>L23</u>	L18 and folding.ab.	2	<u>L23</u>

<u>L22</u>	L17 and (Loop and constant).ab.	0	<u>L22</u>
<u>L21</u>	L20 and (eliminate or replace)	62	<u>L21</u>
<u>L20</u>	L17 and Loop and constant	91	<u>L20</u>
<u>L19</u>	L18 and folding	12	<u>L19</u>
<u>L18</u>	L17 and elimination	76	<u>L18</u>
<u>L17</u>	L3 and (compare)	312	<u>L17</u>
<u>L16</u>	L15 and stub	13	<u>L16</u>
<u>L15</u>	L3 and (dead ADJ code ADJ elimination)	93	<u>L15</u>
<u>L14</u>	L3 and (stub SAME (remove or eliminate))	2	<u>L14</u>
<u>L13</u>	L3 and (stub NEAR (remove or eliminate))	0	<u>L13</u>
<u>L12</u>	L3 and (stub and (remove or eliminate))	26	<u>L12</u>
<u>L11</u>	5671419.pn.	1	<u>L11</u>
<u>L10</u>	L3 and (empty ADJ function)	7	<u>L10</u>
<u>L9</u>	L8 and (type adj analysis)	9	<u>L9</u>
<u>L8</u>	L7 or L6 or L5 or L4	157	<u>L8</u>
<u>L7</u>	L3 and instantiation	89	<u>L7</u>
<u>L6</u>	L4 and object	59	<u>L6</u>
<u>L5</u>	L4 and constraint	15	<u>L5</u>
<u>L4</u>	L3 and (exception adj handler)	74	<u>L4</u>
<u>L3</u>	(717/151 717/152 717/153 717/154 717/155 717/156 717/157 717/158 717/131 717/132 717/133).ccls.	1542	<u>L3</u>
<u>L2</u>	L1 and (exception adj handler).ab.	24	<u>L2</u>
<u>L1</u>	(exception adj handler) AND ((optimization or optimizer or profiler)OR (717/146 717/147 717/148 717/149 717/150 717/151 717/152 717/153 717/154 717/155 717/156 717/157 717/158 717/159 717/129 717/130 717/131 717/132 717/133).ccls.)	361	<u>L1</u>

END OF SEARCH HISTORY

Print Selection

Reset

Cancel

Print

Print First Page

 Section: Page(s): Print Copy: 1

Print? <input type="button" value="All"/> <input type="button" value="Clr"/>	Change? <input type="button" value="All"/> <input type="button" value="Clr"/>	Document ID	Section(s)	Page# or Page Range to Print	Total # of Pages	# of Copies	Database
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6760907	<input type="text" value="all"/>	<input type="text" value="all"/>	12	<input type="text" value="1"/>	USPT
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6591416	<input type="text" value="all"/>	<input type="text" value="all"/>	21	<input type="text" value="1"/>	USPT
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6513156	<input type="text" value="all"/>	<input type="text" value="all"/>	23	<input type="text" value="1"/>	USPT
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	7013459	<input type="text" value="all"/>	<input type="text" value="all"/>	33	<input type="text" value="1"/>	USPT
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6226789	<input type="text" value="all"/>	<input type="text" value="all"/>	* 158	<input type="text" value="1"/>	USPT
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6910205	<input type="text" value="all"/>	<input type="text" value="all"/>	22	<input type="text" value="1"/>	USPT




Note: Print requests for more than 49 pages are denoted by '*' and are in red.

Building

Room

Printer

Freeform Search

Database:	<div style="border: 1px solid black; padding: 2px;"> US Pre-Grant Publication Full-Text Database US Patents Full-Text Database US OCR Full-Text Database EPO Abstracts Database JPO Abstracts Database Derwent World Patents Index IBM Technical Disclosure Bulletins </div>
Term:	<div style="border: 1px solid black; padding: 2px;"> L43 and (alias) <div style="float: right;">    </div> </div>
Display:	<input type="text" value="50"/> Documents in Display Format: <input type="text" value="REV"/> Starting with Number <input type="text" value="1"/>
Generate: <input type="radio"/> Hit List <input checked="" type="radio"/> Hit Count <input type="radio"/> Side by Side <input type="radio"/> Image	

Search History

DATE: Monday, March 03, 2008
[Purge Queries](#)
[Printable Copy](#)
[Create Case](#)

<u>Set</u> <u>Name</u>	<u>Query</u>	<u>Hit</u> <u>Count</u>	<u>Set</u> <u>Name</u> result set
side by side			
<i>DB=USPT; PLUR=NO; OP=OR</i>			
<u>L45</u>	L43 and (alias)	0	<u>L45</u>
<u>L44</u>	L43 and (symbol or namespace)	0	<u>L44</u>
<u>L43</u>	L42 and exception	6	<u>L43</u>
<u>L42</u>	L41 and array	6	<u>L42</u>
<u>L41</u>	L40 and instantiation	6	<u>L41</u>
<u>L40</u>	L39 and call	14	<u>L40</u>
<u>L39</u>	L38 AND (constraint ADJ analysis)	16	<u>L39</u>
<u>L38</u>	717/\$\$\$ccls.	8108	<u>L38</u>
<u>L37</u>	6519765.pn.	1	<u>L37</u>
<u>L36</u>	L35 AND 717/150-159.ccls.	25	<u>L36</u>
<u>L35</u>	L33 or l32	158	<u>L35</u>
<u>L34</u>	L33 and l32	0	<u>L34</u>
<u>L33</u>	native AND type AND object AND (virtual ADJ function)	151	<u>L33</u>
<u>L32</u>	virtual AND function AND constraints AND (sensitive ADJ analysis)	7	<u>L32</u>
native AND type AND object AND virtual AND function AND constraints			

<u>L31</u>	AND (sensitive ADJ analysis)	0	<u>L31</u>
<u>L30</u>	L29 and native.ab.	8	<u>L30</u>
<u>L29</u>	L27 and 717/151-159.ccls.	52	<u>L29</u>
<u>L28</u>	L27 and 717/\$\$\$ccls.	192	<u>L28</u>
<u>L27</u>	L26 and (virtual ADJ function)	453	<u>L27</u>
<u>L26</u>	(source AND Code and optimization) and native (object same type)	290844	<u>L26</u>
<u>L25</u>	L24 and constant	12	<u>L25</u>
<u>L24</u>	L17 and (Loop).ab.	23	<u>L24</u>
<u>L23</u>	L18 and folding.ab.	2	<u>L23</u>
<u>L22</u>	L17 and (Loop and constant).ab.	0	<u>L22</u>
<u>L21</u>	L20 and (eliminate or replace)	62	<u>L21</u>
<u>L20</u>	L17 and Loop and constant	91	<u>L20</u>
<u>L19</u>	L18 and folding	12	<u>L19</u>
<u>L18</u>	L17 and elimination	76	<u>L18</u>
<u>L17</u>	L3 and (compare)	312	<u>L17</u>
<u>L16</u>	L15 and stub	13	<u>L16</u>
<u>L15</u>	L3 and (dead ADJ code ADJ elimination)	93	<u>L15</u>
<u>L14</u>	L3 and (stub SAME (remove or eliminate))	2	<u>L14</u>
<u>L13</u>	L3 and (stub NEAR (remove or eliminate))	0	<u>L13</u>
<u>L12</u>	L3 and (stub and (remove or eliminate))	26	<u>L12</u>
<u>L11</u>	5671419.pn.	1	<u>L11</u>
<u>L10</u>	L3 and (empty ADJ function)	7	<u>L10</u>
<u>L9</u>	L8 and (type adj analysis)	9	<u>L9</u>
<u>L8</u>	L7 or L6 or L5 or L4	157	<u>L8</u>
<u>L7</u>	L3 and instantiation	89	<u>L7</u>
<u>L6</u>	L4 and object	59	<u>L6</u>
<u>L5</u>	L4 and constraint	15	<u>L5</u>
<u>L4</u>	L3 and (exception adj handler)	74	<u>L4</u>
<u>L3</u>	(717/151 717/152 717/153 717/154 717/155 717/156 717/157 717/158 717/131 717/132 717/133).ccls.	1542	<u>L3</u>
<u>L2</u>	L1 and (exception adj handler).ab.	24	<u>L2</u>
<u>L1</u>	(exception adj handler) AND ((optimization or optimizer or profiler)OR (717/146 717/147 717/148 717/149 717/150 717/151 717/152 717/153 717/154 717/155 717/156 717/157 717/158 717/159 717/129 717/130 717/131 717/132 717/133).ccls.)	361	<u>L1</u>

END OF SEARCH HISTORY